

1. Definim un arbre binari genèric així:

```
data Tree a = Empty | Node a (Tree a) (Tree a)
```

Feu que els arbres genèrics siguin instàncies de la classe **Show** i s'escriguin a través del recorregut en inordre dels seus elements, seguint el format dels exemples.

2. Feu que els arbres siguin instància de la classe **Functor** i implementeu una funció `doubleT :: Num a => Tree a -> Tree a` que dobli els valors dels nodes d'un arbre, tot utilitzant el functor d'arbres.

3. Definim un bosc com una llista d'arbres binaris:

```
data Forest a = Forest [Tree a] deriving (Show)
```

Feu que els boscos d'arbres siguin instància de la classe **Functor** i implementeu una funció `doubleF :: Num a => Forest a -> Forest a` que dobli els valors dels nodes dels arbres d'un bosc, tot utilitzant el functor de boscos.

Observació

A l'hora de corregir es tindrà en compte la correcció, senzillesa, elegància i eficiència de la solució proposada.

Exemple d'entrada 1

```
Empty :: Tree Int
Node 1 Empty Empty
(Node 1 (Node 2 Empty Empty) Empty)
(Node 2 Empty (Node 1 Empty Empty))
(Node 1 (Node 2 Empty Empty) (Node 3 Empty Empty))
(Node 1 (Node 2 Empty (Node 3 Empty Empty)) (Node 4 Empty Empty))
```

Exemple de sortida 1

```
()
((),1,())
(((),2,()),1,())
((),2,(((),1,()))
(((),2,()),1,(((),3,()))
(((),2,(((),3,()))),1,(((),4,()))
```

Exemple d'entrada 2

```
doubleT $ Empty :: Tree Int
doubleT $ Node 1 Empty Empty
doubleT $ Node 1 (Node 2 Empty Empty) Empty
doubleT $ Node 2 Empty (Node 1 Empty Empty)
doubleT $ Node 1 (Node 2 Empty Empty) (Node 3 Empty Empty)
doubleT $ Node 1 (Node 2 Empty (Node 3 Empty Empty)) (Node 4 Empty Empty)
```

Exemple de sortida 2

```
()  
((), 2, ())  
(((), 4, ()), 2, ())  
((), 4, ((), 2, ()))  
(((), 4, ()), 2, ((), 6, ()))  
(((), 4, ((), 6, ())), 2, ((), 8, ()))
```

Exemple d'entrada 3

```
doubleF $ Forest []  
doubleF $ Forest [Empty, Node 1 Empty Empty, Node 1 (Node 2 Empty Empty) (Node 3 Empty Empty)]
```

Exemple de sortida 3

```
Forest []  
Forest [(), ((), 2, ()), (((), 4, ()), 2, ((), 6, ()))]
```

Informació del problema

Autor : Jordi Petit, Gerard Escudero

Generació : 2019-10-31 12:29:46

© *Jutge.org*, 2006–2019.

<https://jutge.org>